

NBSIR 74-540

Interactive Graphics on the Sound Laboratory Data Acquisition System

A. James Baroody, Jr.

Mechanics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D. C. 20234

August 1974

Final



U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NBSIR 74-540

**INTERACTIVE GRAPHICS ON THE SOUND
LABORATORY DATA ACQUISITION SYSTEM**

A. James Baroody, Jr.

Mechanics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D. C. 20234

August 1974

Final

U. S. DEPARTMENT OF COMMERCE, Frederick B. Dent, Secretary
NATIONAL BUREAU OF STANDARDS, Richard W. Roberts, Director

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the material or equipment identified is necessarily the best available for the purpose.

PREFACE

This report is intended to assist in the use of the Tektronix 4010-1 graphics display terminal interfaced to the Sound Laboratory Data Acquisition System. The report aims to compile documentation which is peculiar to the Sound Laboratory Data Acquisition System and which is not widely available from other sources.

The author is deeply indebted to Roy Stehle for his substantial contributions to the development of the Access Level Software. The author is also indebted to Will Gallant for his contributions to the modifications of the Tektronix PLOT-10 Terminal Control System.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. FEATURES OF THE INTERACTIVE GRAPHICS SYSTEM	2
2A. The Tektronix 4010-1 Graphics Display Terminal	2
2B. Tektronix PLOT-10 Advanced Graphing II	3
2C. Tektronix PLOT-10 Terminal Control System	3
2D. Access Level Software	3
3. ACCESS LEVEL SOFTWARE	4
3A. TPLOT	4
3B. CURSIS	14
3C. TOUTPT	20
3D. TINPUT	25
3E. PSWMOD	30
4. MODIFICATIONS TO THE TEKTRONIX PLOT-10 TERMINAL CONTROL SYSTEM FOR OPERATION AT 9600 BAUD	34
4A. A listing of the Modifications to TCS	34
REFERENCES	37

1. INTRODUCTION

The objective of laboratory automation is to aid the scientist in the transfer of data from the laboratory measurement to the finished report. To achieve this objective it is necessary for the scientist to properly relate his understanding of scientific principles to the process occurring in the laboratory. To aid the scientist, the computer system must provide information in a compact, descriptive, and understandable form. Computer graphics allows the scientist to communicate conveniently and in his own terms.

This report describes the implementation of an interactive graphics display system on the Sound Laboratory Data Acquisition System. A brief description of the hardware and the software which supports it is presented. A detailed description of computer system-dependent programs required to support this graphics system follows.

2. FEATURES OF THE INTERACTIVE GRAPHICS SYSTEM

Introduction

The key subsystem of the Sound Laboratory Data Acquisition System is an Interdata Model 70 minicomputer. The graphics display, a Tektronix 4010-1 Graphics Display Terminal, is interfaced to the Interdata Model 70 via the Interdata programmable asynchronous line system.

Three levels of software are implemented for support of the Tektronix 4010-1 Graphics Display Terminal under the Interdata Basic Operating System (BOSS) and the Interdata Disc Operating System (DOS). The software implemented includes the Tektronix PLOT-10 Advanced Graphing II, the Tektronix PLOT-10 Terminal Control System, and the Access Level Software.

2A. The Tektronix 4010-1 Graphics Display Terminal

The 4010-1 terminal consists of a keyboard and a viewing screen. The viewing screen incorporates a direct-view storage tube, which in outward appearance behaves like a CRT with an extremely long-persistence phosphor. Two separate principles are involved in the creation of a display. The first is the storage of an image on a grid mounted just behind the screen. This image is then transferred to the screen by means of a flood of electrons.

The 4010-1 operates in three modes as follows:

- 1) Alphanumeric Modes Characters entered via the keyboard or received from the Interdata Model 70 are displayed using an internal character generator.
- 2) Graphic Plot Mode. In the graphic plot mode, the 4010-1 divides the display into a matrix of 1024 addressable points on both the horizontal (X) and the vertical (Y) axis. Only 780 of these points are visible on the Y axis. Using an internal vector generator, the 4010-1 draws vectors under control of the Interdata Model 70.
- 3) Graphic Input Mode. In this mode the Interdata Model 70 activates the cross-hair cursor. The cursor can be positioned to the desired intersect point by the operator using the thumbwheel cursor controls. When positioned at the appropriate location, the cross-hair coordinates are transmitted to the Interdata Model 70 by an operator command.

2B. Tektronix PLOT-10 Advanced Graphing II (AG-II)

The Tektronix Advanced Graphing II (AG-II) package is a high-level graphics language written in FORTRAN IV (ANSI X3.9-1966 compatible). This language permits the user with a basic understanding of graphics, in general, and of the Tektronix 4010-1 to perform plotting on the Tektronix 4010-1. All that is required is the creation of a program containing CALL's to the appropriate routines. Routines are provided to perform functions such as data manipulation, scaling, axis generation, labeling, and plotting. A minimum of three CALL's are required to generate a full screen plot. For further details, see the PLOT-10 Advanced Graphing II User's Manual.

Note: The Terminal Control System (Section 2C) is required for the operation of Advanced Graphing II.

2C. Tektronix PLOT-10 Terminal Control System (TCS)

The Tektronix PLOT-10 Terminal Control System (TCS) is a set of FORTRAN IV (ANSI X3.9-1966 compatible) terminal control primitives for the Tektronix 4010-1. These primitives, through subroutine calls, provide the user with full control of all features of the Tektronix 4010-1. TCS provides the user with graphic functions such as vector generation, windowing, clipping and rotation. For further details see the Tektronix Terminal Control System User's Manual.

Note: The Access Level Software (Section 2D) is required for the operation of TCS.

2D. Access Level Software (ALS)

The Access Level Software (ALS) implements four primitives to provide the FORTRAN or assembly language programmer with support for graphic input and output via the Interdata multiplexor bus. The cursor is supported in addition to the terminal. The software includes basic routines for plotting, graphic input, the input of any ASCII character, and the output of any ASCII character.

3. ACCESS LEVEL SOFTWARE

The Access Level Software (ALS) has been written to facilitate the use of the 4010-1 with the Interdata Model 70. ALS includes four assembly language routines. These four routines perform the basic functions of the Tektronix 4002A Access Level Software, but have been rewritten to conform to the conventions of the Interdata OS Library Loader. For details on the 4002A Access Level Software refer to the Minicomputer-4002A Interdata 3 and 4 User's Manual.

All of the ALS routines are written in assembly language. If calls are made from other assembly language routines, the entries to the required routines should be defined with an EXTRN statement, i.e., EXTRN, TINPUT, TOUPUT, TPLOT, CURSIS. The ALS routines are also FORTRAN-callable, but not re-entrant. Registers 13, 14, and 15 are saved, used, and then restored.

All of the routines assume the 4010-1 graphics display is interfaced to the Interdata PALS system port at address X'33'.

The ALS routines are written using sense status loops and privileged instructions. As a result, interrupts should be disabled as should the Protect Mode. To insure proper execution using all versions of the Basic Operating System (BOSS) and the Disc Operating System (DOS), a program called PSWMOD is included as an assistance to implementing these functions.

3A. TPLOT

Function: Performs plotting as specified in either a linear interpolation or point plotting mode.

FORTRAN Usage: CALL TPLOT(MODE, IX, IY)

Assembly Usage: BAL 15, TPLOT
DC 8
(ADDRESS OF MODE)
(ADDRESS OF IX)
(ADDRESS OF IY)
(RETURN HERE)

Description: MODE defines the type of plotting to be performed.

If MODE = 0 Initialize and dark vector to IX, IY
 > 0 Bright vector to IX, IY
 < 0 Point plot at IX, IY

IX and IY are integers in the range 0,1023. The plotting position is determined by the parameters IX and IY. If IX or IY is negative, the value used in plotting will be zero. If IX and IY is greater than 1023, a value of 1023 will be used for plotting.

Dark Vector If MODE = 0, the terminal is set to linear interpolation mode and the graphics cursor is initialized to the specified (IX,IY) position.

Bright Vector If MODE > 0, the terminal is assumed to be in linear interpolation mode and a vector is drawn from the previous (IX,IY) coordinate to the specified (IX,IY) coordinate. A call to TPL0T in dark vector mode must precede plotting in bright vector mode in order to effect proper initialization of the 4010-1.

Point Plot If MODE < 0, an intensified point is plotted at the specified (IX,IY) position.

For linear interpolation and point plot modes, the coordinate values (IX,IY) are decoded into four 5-bit bytes. The four resulting bytes are then sent to the terminal through TOUTPT in this order: High Order IY, Low Order IY, High Order IX, and Low Order IX with flag bits 6 and 7 configured to Table 1. Positions taken by the 5 coordinate value bits are represented by W.

	Control Bits			Data			
High Order IY	0	1	W	W	W	W	W
Low Order IY	1	1	W	W	W	W	W
High Order IX	0	1	W	W	W	W	W
Low Order IX	1	0	W	W	W	W	W

Table 1

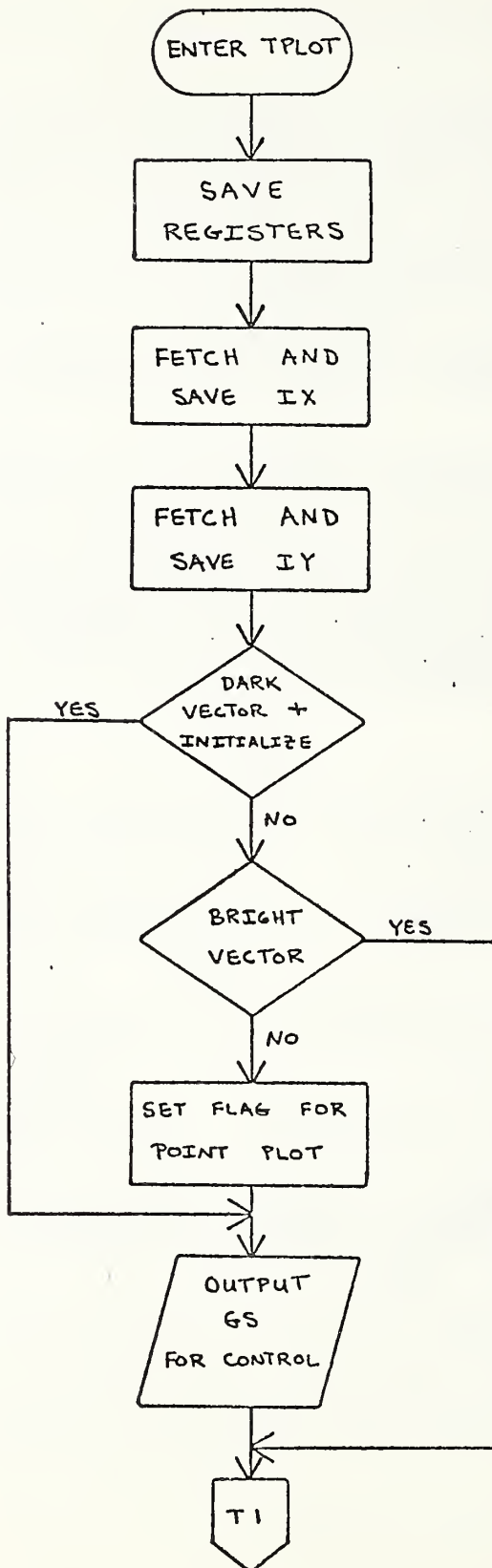
The point (100,200) would be decoded as:

$$200_{10} = \overset{\text{IY}}{(0011001000)}_2 ; \quad 100_{10} = \overset{\text{IX}}{(0001100100)}$$

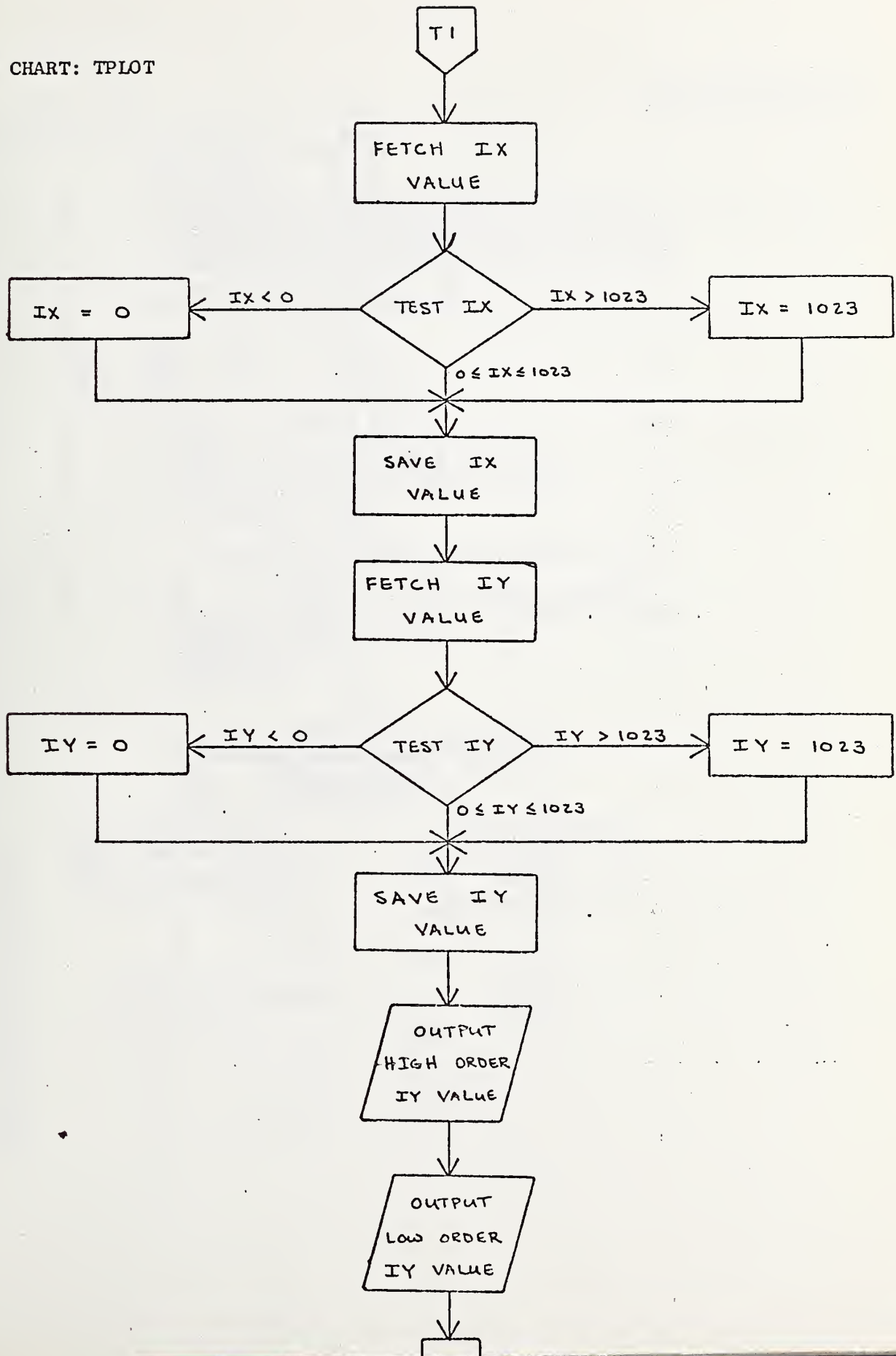
High Order IY = 0100110 High Order IX = 0100011
Low Order IY = 1101000 Low Order IX = 1000100

Subroutines Called: TOUTPT

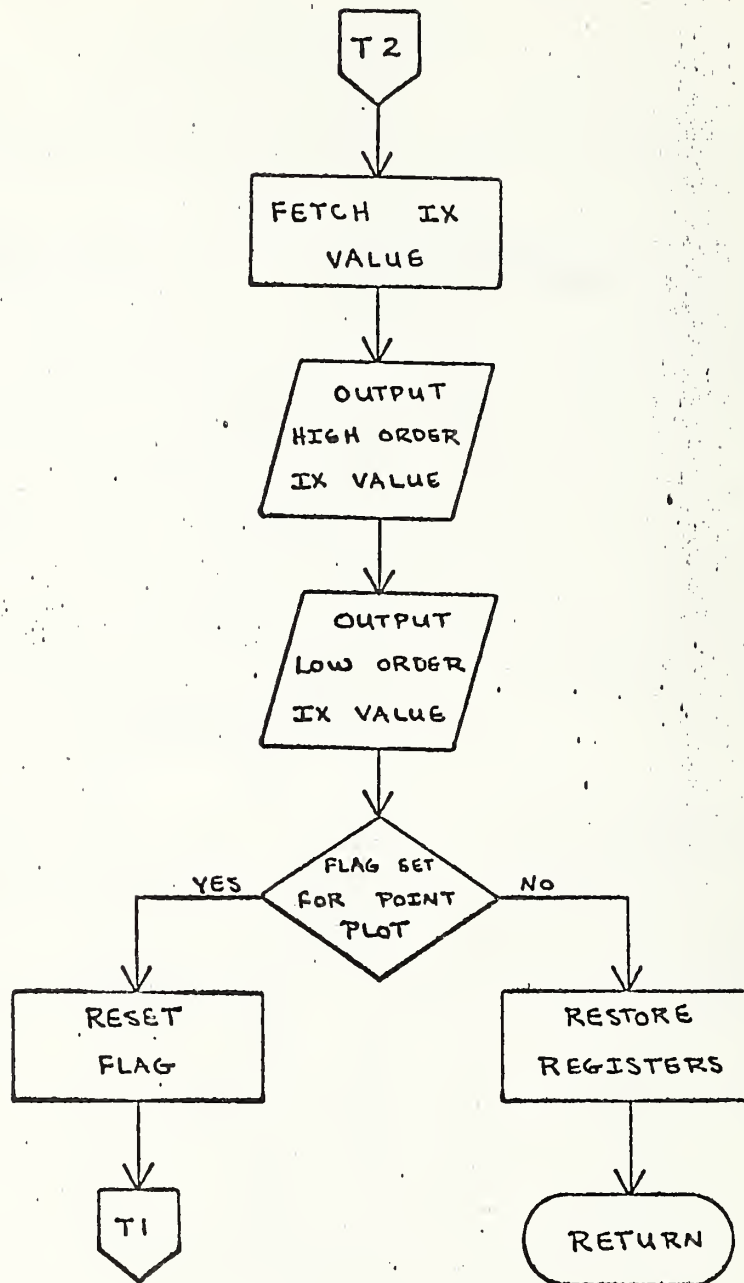
FLOW CHART: T PLOT



FLOW CHART: TPLOT



FLOW CHART: TPLOT




```

TPlot(MODE, IX, IY) SUBROUTINE
0000R      ENTRY TPlot, IPlot
0000R      EXTRN CHOUT

*          THE FOLLOWING PACKAGE IS DESIGNED TO
*          OPERATE ON THE INTERDATA MODEL 70 COMPUTER
*          INTERFACED TO THE TEKTRONIX 4010-1 USING THE
*          PROGRAMMABLE ASYNCHRONOUS LINE SYSTEM (PALS).
*
*          THIS PACKAGE USES REGISTERS 13, 14, AND 15
*
0000D      R13      EQU      13
0000E      R14      EQU      14
0000F      RTN      EQU      15
*
*          TPlot
*          THIS ROUTINE IS CALLED TO PLOT IN
*          VECTOR, POINT, OR INCREMENTAL PLOT
*          MODE, DEPENDING ON THE VALUE OF MODE
*          AS DESCRIBED BELOW.
*
*          TO CALL          CALL TPlot(MODE, IX, IY)
*          BAL      RTN, TPlot
*          DC      8          IX, IY ARE INTEGERS
*                          IN RANGE 0, 1023
*          (ADDRESS OF MODE)
*          (ADDRESS OF X)
*          (ADDRESS OF Y)
*          (RETURN HERE)
*
*          IF
*          MODE = 0 INITIALIZE AND DARK VECTOR
*                  TO X, Y
*          MODE > 0 BRIGHT VECTOR TO X, Y
*          MODE < 0 POINT PLOT AT X, Y
*
00009R 2300      IPlot  BFFS  0, 0          DUMMY ENTRY FOR DIFF PROG
0002R  D000      TPlot  STM   R13, RSAVE      SAVE REGISTERS
          00FAR
0006R  40F0      STH   RTN, RADD          SAVE RETN ADD
          00F2R
000AR  48EF      LH    R14, 4(RTN)        ADD OF X
          0004
000ER  48DE      LH    R13, 0(R14)        X
          0000
0012R  40D0      STH   R13, TPTX          SAVE X
          00F4R
0016R  48EF      LH    R14, 6(RTN)        ADD OF Y
          0006
001AR  48DE      LH    R13, 0(R14)        Y
          0000

```

TPlot(MODE, IX, IY)		SUBROUTINE		
001ER	40D0	STH	R13, TPTY	SAVE Y
	00F6R			
0022R	48EF	LH	R14, 2(RTN)	ADD OF MODE
	0002			
0026R	48DE	LH	R13, 0(R14)	MODE
	0000			
002AR	4330	BZ	TPTDV	ZERO GO INIT AND DV
	00E8R			
002ER	4220	BP	TPTNRM	+ NORMAL
	0044R			
0032R	4300	B	TPTDV	(POINT PLOT)
	00E8R			
0036R	48F0	PLOT LH	RTN, RADD	RETURN ADDRESS
	00F2R			
003AR	48EF	LH	R14, 2(RTN)	MODE ADDRESS
	0002			
003ER	24D1	LIS	R13, 1	
0040R	40DE	STH	R13, 0(R14)	CHANGE MODE=1
	0000			
0044R	48D0	TPTNRM LH	R13, TPTX	GET X
	00F4R			
0048R	4310	BNM	TPT10	JUMP IF POSITIVE
	0050R			
004CR	C8D0	LHI	R13, 0	SET TO EDGE FOR NEG X
	0000			
0050R	C5D0	TPT10 CLHI	R13, 1024	CHECK BOUNDS
	0400			
0054R	4280	BL	TPT20	JUMP IF OK
	005CR			
0058R	C8D0	LHI	R13, 1023	SET TO EDGE OF SCREEN
	03FF			
005CR	40D0	TPT20 STH	R13, TPTX	SAVE NEW X
	00F4R			
0060R	48D0	LH	R13, TPTY	GET Y
	00F6R			
0064R	4310	BNM	TPT30	
	006CR			
0068R	C8D0	LHI	R13, 0	SET TO LOW EDGE
	0000			
006CR	C5D0	TPT30 CLHI	R13, 1024	CHECK BOUNDS
	0400			
0070R	4280	BL	TPT40	JUMP IF OK
	0078R			
0074R	C8D0	LHI	R13, 1023	SET TO EDGE OF FIELD
	03FF			
0078R	40D0	TPT40 STH	R13, TPTY	SAVE Y
	00F6R			
007CR	CCD0	SRHL	R13, 5	GET UPPER 5 BITS

TPLOT(MODE, IX, IY) SUBROUTINE

0005			
0080R C6D0	OHI	R13,X'20'	PUT IN HI Y TAG
0020			
0084R 40D0	STH	R13,TEMP	SAVE FOR OUTPUT
00F8R			
0088R 41F0	BAL	RTN,CHOUT	OUTPUT HI Y
0000F			
008CR 00F8R	DC	A(TEMP)	
008ER 48D0	LH	R13,TPTY	GET Y
00F6R			
0092R C4D0	NHI	R13,X'1F'	MASK TO LOW 5
001F			
0096R C6D0	OHI	R13,X'60'	LOW Y TAG
0060			
009AR 40D0	STH	R13,TEMP	
00F8R			
009ER 41F0	BAL	RTN,CHOUT	OUTPUT LOW Y
008AR			
00A2R 00F8R	DC	A(TEMP)	
00A4R 48D0	LH	R13,TPTX	GET X
00F4R			
00A8R C4D0	SRHL	R13,5	GET HIGH 5
0005			
00ACR C6D0	OHI	R13,X'20'	SET IN HIGH TAG
0020			
00B0R 40D0	STH	R13,TEMP	
00F8R			
00B4R 41F0	BAL	RTN,CHOUT	OUTPUT HIGH X
00A0R			
00B8R 00F8R	DC	A(TEMP)	
00BAR 48D0	LH	R13,TPTX	
00F4R			
00BER C4D0	NHI	R13,X'1F'	MASK TO LOW 5
001F			
00C2R C6D0	OHI	R13,X'40'	LOW X TAG
0040			
00C6R 40D0	STH	R13,TEMP	
00F8R			
00CAR 41F0	BAL	RTN,CHOUT	OUTPUT LOW X
00B6R			
00CER 00F8R	DC	A(TEMP)	
00D0R 48F0	LH	RTN,RADD	GET RETURN ADDRESS
00F2R			
00D4R 48EF	LH	R14,2(RTN)	ADD OF MODE
0002			
00D8R 48EE	LH	R14,0(R14)	MODE
0000			
00DCR 4210	BM	PPL0T	BRANCH FOR POINT PLOT

TPTRTN

.TPLOT(MODE,IX,IY) SUBROUTINE

0036R				
00E0R D1D0		LM	R13,RSAVE	RESTORE REGISTERS
00FAR				
00E4R 430F		B	8(RTN)	EXIT
0008				
00E8R 41F0	TPTDV	BAL	RTN,CHOUT	GS TO START
00CCR				
00ECR 0100R		DC	A(GS)	
00EER 4300		B	TPTNRM	
0044R				
00F2R	RADD	DS	2	
00F4R	TPTX	DS	2	
00F6R	TPTY	DS	2	
00F8R	TEMP	DS	2	
00FAR	RSAVE	DS	6	REGISTER STORAGE AREA
0100R 001D	GS	DC	X'1D'	GRAPHIC MODE
0102R		END		

TPLOT(MODE,IX,IY) SUBROUTINE
NO ERRORS

* CHOUT	00EAR
GS	0100R
* IPLOT	0000R
PPLOT	0036R
R13	000D
R14	000E
RAID	00F2R
RSAVE	00FAR
RTI	000F
TEMP	00F0R
* TPLOT	0002R
TPT10	0050R
TPT20	005CR
TPT30	006CR
TPT40	0078R
TPTDV	00E8R
TPTHRM	0044R
TPTRTN	00D0R
TPTX	00F4R
TPTY	00F6R

3B. CURSIS

Function: Enable input of graphic data by reading the cursor position and a keyboard character

FORTRAN Usage: CALL CURSIS(ICHAR,IX,IY)

Assembly Usage: BAL 15,CURSIS
DC 8
(ADDRESS OF ICHAR)
(ADDRESS OF IX)
(ADDRESS OF IY)
(RETURN HERE)

Description: ICHAR is the decimal equivalent of the first keyboard character struck following the enabling of the cursor

IX is the coordinate of the vertical crosshair when the character was typed (abscissa)

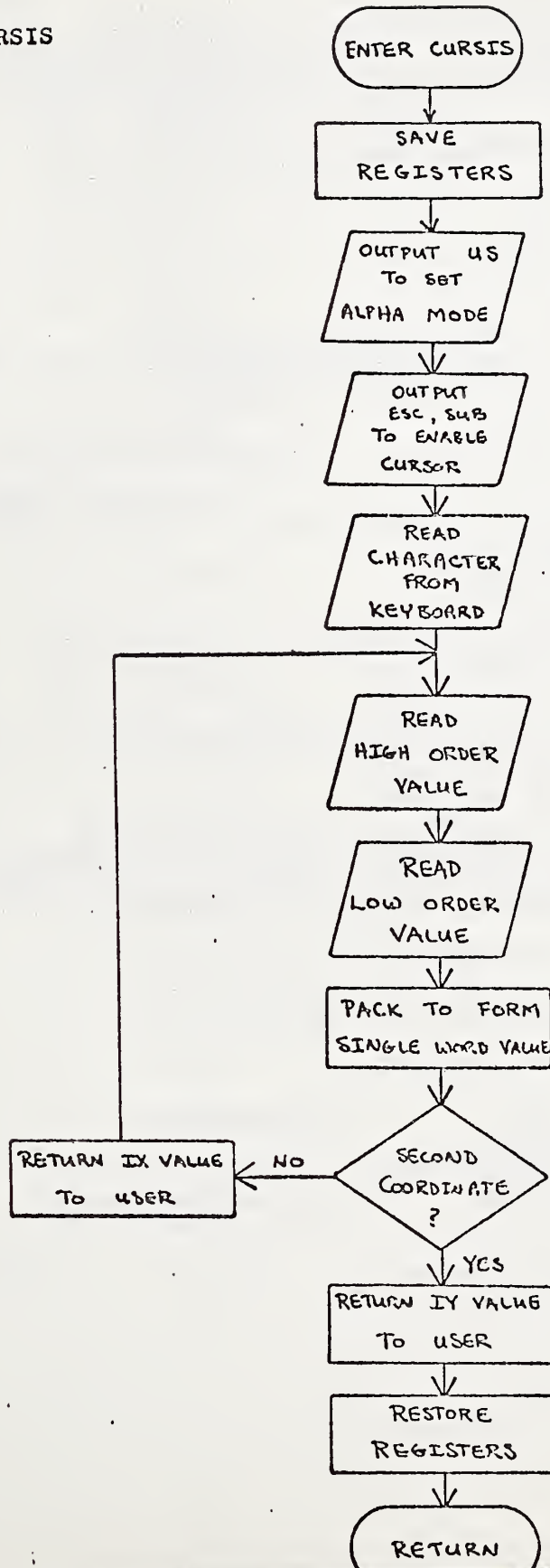
IY is the coordinate of the horizontal crosshair when the character was typed (ordinate)

CURSIS enables the graphics cursor by outputting the ASCII characters US,ESC, and SUB. With the cursor enabled, CURSIS waits for the next keyboard character to be struck which will allow CURSIS to read the keyboard character and four additional characters representative of the cursor's position. Parity is stripped from all characters. The keyboard character is returned as the integer ICHAR. The next four characters contain coordinate information in the 5 least significant bits of each character. These represent High Order IX, Low Order IX, High Order IY, and Low Order IY.

Subroutines Called: TINPUT, TOUTPT

Notes: CURSIS expects only five ASCII characters back from the terminal. If the Graphic Input Terminators on the terminal board TC-2 append a carriage return or a carriage return and EOT, these will be ignored if the terminal is hard-wired through the teletype port. If character transmission is at a slow rate (e.g. 300 baud), the subroutine TINPUT may be necessary to accept these characters.

FLOW CHART: CURSIS



CURSIS(ICHAR,IX,IY) SUBROUTINE

```

0000R      ENTRY CURSIS
0000R      EXTRN CHOUT,CHIN1,CHIN2

*          THE FOLLOWING PACKAGE IS DESIGNED TO
*          OPERATE ON THE INTERDATA MODEL 70 COMPUTER
*          INTERFACED TO THE TEKTRONIX 4010-1 USING THE
*          PROGRAMMABLE ASYNCHRONOUS LINE SYSTEM (PALS).
*
*          THIS PACKAGE USES REGISTERS 13,14, AND 15
*
0000D      R13      EQJ      13
0000E      R14      EQJ      14
0000F      RTN      EQU      15
*
*          CURSIS
*          THIS ROUTINE IS USED FOR GRAPHICS
*          INPUT IF THE INTERACTIVE GRAPHIC
*          UNIT AND JOY STICK ARE AVAILABLE
*
*          TO CALL          CALL CURSIS(ICHAR,IX,IY)
*          BAL      RTN,CURSIS
*          DC      8          IX,IY ARE INTEGERS
*          (ADDRESS OF CHARACTER WORD)    IN RANGE 0,1023
*          (ADDRESS OF X)
*          (ADDRESS OF Y)
*          (RETURN HERE)
*
*          THE CHARACTER IS STORED IN THE RIGHT BYTE
*          OF THE WORD
*
*          CURSIS FIRST SETS THE 4010-1 TO ALPHA
*          MODE THEN TO GRAPHICS INPUT MODE WHICH
*          TURNS ON THE CURSOR.
*
*          WHEN A CHARACTER IS TYPED THE CHARACTER
*          IS RETURNED AS WELL AS THE X AND Y COORDINATES
*          OF THE CURSOR.  ALL CHARACTERS AFTER LOW X ARE
*          NOT READ (AND, GENERALLY, NEED NOT BE).
*
0000R 0000D      CURSIS STM      R13,RSAVE      SAVE REGISTERS
0000R 0000E      0000R      STH      RTN,RADD      SAVE RETN ADD
0000R 0000F      0000R      BAL      RTN,CHOUT
0000R 0000R      0000R      DC      A(US)          RESET TERMINAL
0000R 0000R      0000R      BAL      RTN,CHOUT
0000R 0000R      0000R      DC      A(CURS)        TURN ON CURSOR (ESC)
0000R 0000R      0000R

```


CURSIS(ICHAR, IX, IY) SUBROUTINE

0014R	41F0	BAL	RTN,CHOUT	
	0010R			
0018R	0077R	DC	A(CURS-1)	(SUB)
001AR	41F0	BAL	RTN,CHIN1	GET THE CHAR
	0000F			
001ER	C8E0	LHI	R14,3	LOOP COUNTER
	0003			
0022R	40E0	STH	R14,TPTY	SAVE IN COUNTER
	007ER			
0026R	4300	B	CUR20	GO TO LOOP
	0048R			
002AR	41F0	CUR10 BAL	RTN,CHIN2	GET HI VAL
	0000F			
002ER	40D0	STH	R13,TEMP	SAVE HI VALUE
	0080R			
0032R	41F0	BAL	RTN,CHIN2	LOW VAL
	002CR			
0036R	C4D0	NHI	R13,X'1F'	LEAVE LOWER 5 ABITS
	001F			
003AR	48E0	LH	R14,TEMP	GET HIGH VALUE
	0090R			
003ER	CDE0	SLHL	R14,5	SHIFT UP 5 BITS
	0005			
0042R	06DE	OHR	R13,R14	PUT TOGETHER
0044R	C4D0	NHI	R13,X'3FF'	MASK TO 10 BITS
	03FF			
0048R	48F0	CUR20 LH	RTN,RADD	GET ADDRESS
	007CR			
004CR	48EF	LH	R14,2(RTN)	PARAMETER ADDR
	0002			
0050R	40DE	STH	R13,0(R14)	STORE DATA
	0000			
0054R	CAFO	AHI	RTN,2	COUNT UP ADDRESS
	0002			
0058R	48D0	LH	R13,TPTY	GET COUNTER
	007ER			
005CR	CB00	SHI	R13,1	COUNT DOWN LOOP
	0001			
0060R	4330	BZ.	CURRTH	RETURN
	0070R			
0064R	40F0	STH	RTN,RADD	
	007CR			
0068R	40D0	STH	R13,TPTY	STORE COUNTER TOO
	007ER			
006CR	4300	B	CUR10	
	002AR			
0070R	D1D0	CURRTH LM	R13,RSAVE	RESTORE REGISTERS
	0082R			

CURSI (ICAR, IX, IY) SUBROUTINE

0074R	430F	B	8(RTN)	EXIT
	0008			
0078R	1A1B	CURS	DC X'1A1B'	CURSOR (ESC, SUB)
007AR	001F	US	DC X'1F'	RESET CONTROL SHIFT 0
007CR		RADD	DS 2	
007ER		TPTY	DS 2	
0080R		TEMP	DS 2	
0082R		RSAVE	DS 6	REGISTER STORAGE AREA
0088R		END		

CURSIG(ICHAR, IX, IY) SUBROUTINE

END

0028R

* CHIN1 001CR

* CHIN2 0034R

* CHOUT 0016R

CUP10 002AR

CUP20 0048R

CURRTN 0070R

CURS 0078R

* CURSIG 0000R

R13 000D

R14 000E

RADD 007CR

RSAVE 0082R

RTN 000F

TEMP 0080R

TPTY 007ER

US 007AR

3C. TOUTPT

Function: Outputs an 8-bit character to the 4010.

FORTTRAN Usage: CALL TOUTPT(ICHAR)

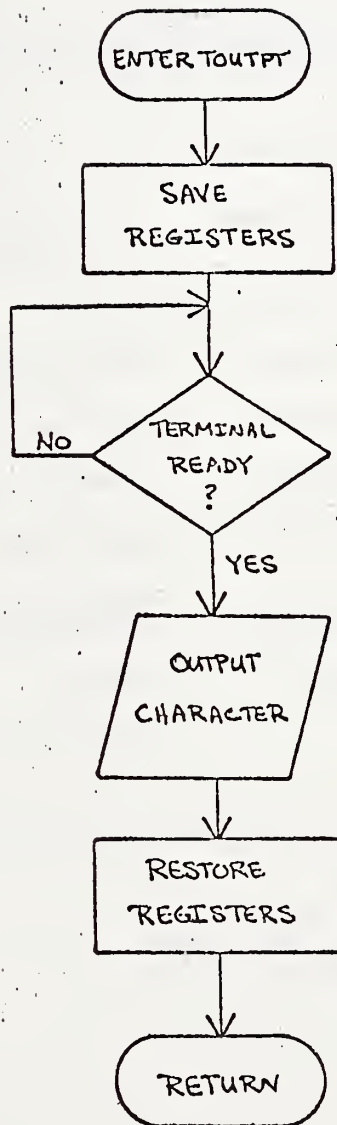
Assembly Usage: BAL 15, TOUTPT
DC 2
(ADDRESS OF ICHAR)
(RETURN HERE)

Description: ICHAR is the decimal equivalent of the ASCII character to be sent to the 4010.

TOUTPT will output the 8-bit character at location ICHAR. Normally only the lower-order 7 bits are used for the 128 ASCII characters.

Notes: The assembly language programmer is given the option of calling the routine CHOUT to retain some similarity to earlier Tektronix software. The programmer is referred to the program listing for more information. If used, he must LINK TOUTPT rather than using EDIT.

FLOW CHART: TOUTPT



TOUTPT(ICHAR) SUBROUTINE

```

0000R      ENTRY TOUTPT,CHOUT
          *      THE FOLLOWING PACKAGE IS DESIGNED TO
          *      OPERATE ON THE INTERDATA MODEL 3,4,5, OR 70
          *      COMPUTER.
          *
          *      THIS PACKAGE USES REGISTERS 13,14, AND 15
          *
0000D      R13      EQU      13
0000E      R14      EQU      14
0000F      RTN      EQU      15
00009      BUSY     EQU      9              BUSY & DEV. UNAVAIL.
          *
          *      TOUTPT
          *      THIS ROUTINE IS CALLED TO OUTPUT
          *      AN ASCII CHARACTER TO THE 4010-1
          *      GRAPHIC COMPUTER TERMINAL (ADDRESS X'33').
          *
          *      TO CALL
          *      BAL      RTN,TOUTPT      CALL TOUTPT(ICHAR)
          *      DC        2
          *      (ADDRESS OF CHAR)      ICHAR IS DEC. VALUE
          *      (RETURN HERE)          OF CHARACTER
          *
          *      REGISTERS 13-15 ARE RESTORED
          *
0000R 0000      TOUTPT STM      R13,SAVREG      SAVE REGISTERS
          001AR
0004R 48EF      LH        R14,2(RTN)          GET CHAR ADDR
          0002
0008R 40E0      STH      R14,CHADD          TRANSFER CHAR ADDR
          0010R
000CR 41F0      BAL      RTN,CHOUT          OUTPUT CHAR
          0020R
          CHADD   DS        2              CHAR ADDR
0010R          LM        R13,SAVREG          RESTORE REGISTERS
0012R D1D0
          001AR
0016R 430F      B        4(RTN)            RETURN
          0004
001AR          SAVREG   DS        6              REGISTER SAVE AREA
          *
          *      CHOUT
          *      CHARACTER OUTPUT SUBROUTINE FROM ASSEMBLY LA
          *
          *      TO CALL
          *      BAL      RTN,CHOUT
          *      (ADDRESS OF DATA WORD)
          *      (RETURN HERE)
          *

```

TOUTPT(ICHAR) SUBROUTINE

* THE CHARACTER SHOULD BE IN THE RIGHT BYTE
*
* REGISTERS 13-15 ARE NOT RESTORED

0020R	C8D0	CHOUT	LHI	R13,DEV	GET DEVICE CODE
	0044R				
0024R	DED0		OC	R13,PRATE	SEND RATE INFO
	0046R				
0028R	DED0		OC	R13,SNDCMD	SEND OUT MODE
	0047R				
002CR	9DDE	CHOUT1	SSR	R13,R14	STATUS TO R14
002ER	4290		BTC	BUSY,CHOUT1	JUMP IF BUSY
	002CR				
0032R	48EF		LH	R14,0(RTN)	GET ADDR OF CHAR
	0000				
0036R	DADE		WD	R13,1(R14)	WRITE LOW BYTE
	0001				
003AR	9DDE	CHOUT2	SSR	R13,R14	R14 GETS STATUS
003CR	4290		BTC	BUSY,CHOUT2	
	003AR				
0040R	430F		B	2(RTN)	RETURN
	0002				
0044R	0033	DEV	DC	X'33'	
0046R	F0	PRATE	DB	X'F0'	
0047R	00A3	SNDCMD	DC	X'A3'	
0049R			END		

TOUTPT(ICHAR) SUBROUTINE

D ERRORS

BUSY	0009
CHADD	0010R
CHOUT	0020R
CHOUT1	002CR
CHOUT2	003AR
DEV	0044R
PRATE	0046R
R13	000D
R14	000E
RTH	000F
SAVREG	001AR
SHDCMD	0047R
TOUTPT	0000R

3D. TINPUT

Function: Inputs an 8-bit ASCII character from the 4010.

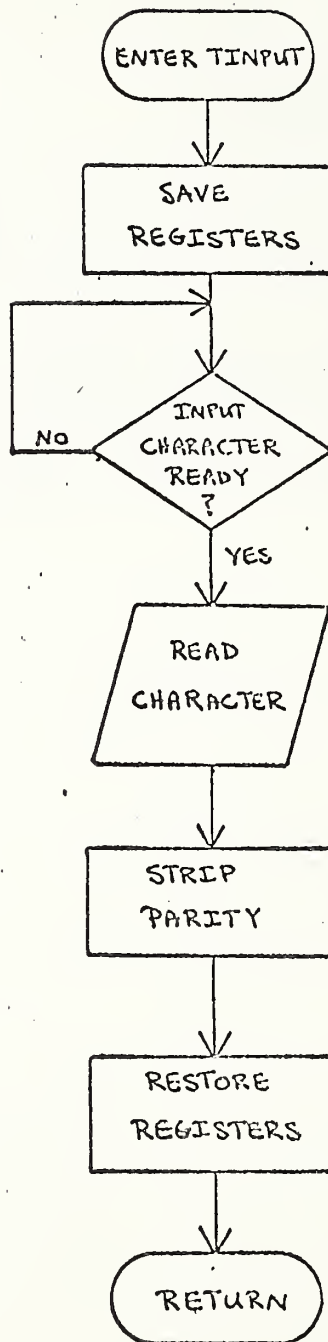
FORTRAN Usage: CALL TINPUT(ICHAR)

Assembly Usage: BAL 15, TINPUT
DC 2
(ADDRESS OF ICHAR)
(RETURN HERE)

Description: TINPUT will receive any of the 128₁₀ ASCII characters generated by the terminal and store its value in ICHAR. The parity bit is stripped from the character.

Notes: The assembly language programmer is given the option of calling the routines CHIN and CHIN1 to retain some similarity to earlier Tektronix software. The programmer is referred to the program listing for more information. If used, he must LINK TINPUT rather than using EDIT.

FLOW CHART: TINPUT



TINPUT(ICHAR) SUBROUTINE

```

0000R      ENTRY TINPUT,CHIN,CHIN1,CHIN2
          *      THE FOLLOWING PACKAGE IS DESIGNED TO
          *      OPERATE ON THE INTERDATA MODEL 3.4.5, OR 70
          *      COMPUTER.
          *
          *      THIS PACKAGE USES REGISTERS 13,14, AND 15
          *
0000D      R13      EQU      13
0000E      R14      EQU      14
0000F      RTN      EQU      15
00009      BUSY     EQU      9              BUSY & DEV. UNAVAIL.
          *
          *      TINPUT
          *      THIS ROUTINE IS CALLED TO INPUT
          *      AN ASCII CHARACTER FROM THE 4010-1
          *      GRAPHIC COMPUTER TERMINAL (ADDRESS X'33').
          *
          *      TO CALL
          *      BAL      RTN,TINPUT          CALL TINPUT(ICHAR)
          *      DC       2
          *      (ADDRESS OF CHARACTER)      ICHAR IS DEC. VALUE
          *      (RETURN HERE)              OF INPUT CHARACTER
          *
          *      REGISTERS 13-15 ARE RESTORED
          *
0000R 2300  CHIN     BFFS   0,0              DUMMY ENTRY FOR DIFF. PROG
0002R 0000  TINPUT   STM    R13,REGSAV      SAVE REGISTERS
          001ER
0006R 41F0          BAL    RTN,CHIN1        GET CHARACTER
          0024R
000AR 48F0          LH     RTN,REGSAV+4     GET RETURN ADDR
          0022R
0000R 48EF          LH     R14,2(RTN)       GET CHAR ADDR
          0002
0012R 40DE          STH    R13,0(R14)       STORE CHARACTER
          0000
0016R D1D0          LM     R13,REGSAV       RESTORE REGISTERS
          001ER
001AR 430F          B      4(RTN)          RETURN
          0004
001ER      REGSAV   DS     6              REGISTER SAVE AREA
          *
          *      CHIN1
          *      SUBROUTINE CALL FROM ASSEMBLY LANGUAGE
          *
          *      TO CALL
          *      BAL      RTN,CHIN1
          *      (RETURN HERE)

```

TINPUT(ICHAR) SUBROUTINE

*

*

CHARACTER IS RETURNED IN LOWER BYTE OF REG 13.

*

REGISTERS 13-15 ARE NOT RESTORED

*

0024R	C8E0	CHIN1	LHI	R14,DEV	LOAD DEVICE ADDRESS
	0044R				
0028R	DEE0		OC	R14,PRATE	SET UP RATE
	0046R				
002CR	DEE0		OC	R14,RCVCMD	SEND RECEIVE MODE
	0047R				
0030R	C8E0	CHIN2	LHI	R14,DEV	LOAD DEVICE ADDRESS
	0044R				
0034R	9DED		SSR	R14,R13	SENSE STATUS
0036R	4290		BTC	BUSY,CHIN2	LOOP IF BUSY OR UNAVAIL
	0030R				
003AR	9BED		RDR	R14,R13	READ CHAR
003CR	C4D0		NHI	R13,X'7F'	STRIP PARITY
	007F				
0040R	430F		B	0(RTN)	RETURN
	0000				
0044R	0033	DEV	DC	X'33'	
0046R	F0	PRATE	DB	X'F0'	
0047R	A1	RCVCMD	DB	X'A1'	
0048R			END		

TINPUT(ICHAR) SUBROUTINE

NO ERRORS

BUSY	0009
* CHII	0000R
* CHIN1	0024R
* CHIN2	0030R
DEV	0044R
PRATE	0046R
R13	000D
R14	000E
RCVCMD	0047R
REGSAV	001ER
RTN	000F
* TINPUT	0002R

3E. PSWMOD

Function: Enables the FORTRAN Programmer to change the current PSW in real time.

Description: FORTRAN Call Statement: CALL ENIOI
This entry sets the external interrupt bit of the current PSW.

FORTTRAN Call Statement: CALL DISIOI
This entry resets the external interrupt bit of the current PSW.

FORTTRAN Call Statement: CALL ENPM
This entry sets the protect mode bit of the current PSW.

FORTTRAN Call Statement: CALL DISPM
This entry resets the protect mode bit of the PSW.

Note: This routine modifies the current PSW via a SVC 2 type 4, the Set Status Supervisor Call. Both the Basic Operating System and the Disc Operating System require modification to permit the user to perform a Set Status Supervisor Call which disables the protect mode. The reader is referred to the Basic Operating System Reference Manual and the Disc Operating System Reference Manual for further details.

PSW STATUS MODIFIER

PAGE 1

*
 *THIS PROGRAM CONTAINS FORTRAN-CALLABLE ENTRY POINTS
 *THAT MODIFY THE BIT PATTERN IN THE CURRENT STATUS
 *PSW AS FOLLOWS:

* CALL ENIOI - ENABLE EXTERNAL INTERRUPTS
 * CALL DISIOI - DISABLES EXTERNAL INTERRUPTS
 * CALL ENPM - ENABLES PROTECT MODE
 * CALL DISPM - DISABLES PROTECT MODE
 *

CHANGE THE FOLLOWING BOSS LOCATIONS (F03-019R01)

* LOC. FROM	TO
* 1E0 NHI ADR.X'3D0F'	NHI ADR.X'7500'
* 1E4 OHI ADR.X'2900'	OHI ADR.X'2800'

*

*

000F RTN EQU 15

*

0000R ENTRY ENIOI,DISIOI,ENPM,DISPM

*

0000R 40F0 ENIOI STH RTN,RSAVE SAVE R15

0064R

0004R E110 SVC 1,WAIT DUMMY CALL

0062R

0008R 48F0 LH RTN,X'96' GET CURRENT PSW STATUS

0096

000CR C6F0 OHI RTN,X'4000' ADD ENABLE IO BIT

4000

0010R 40F0 FINIS STH RTN,STAT+2 STORE SVC 2 CALL

0060R

0014R E120 SVC 2,STAT GET PSW STATUS

005ER

0018R 48F0 LH RTN,RSAVE RESTORE R15

0064R

001CR 4AFF AH RTN,0(RTN) INC FOR NO. OF ARGS

0000

0020R 030F BR RTN RETURN

*

0022R 40F0 DISIOI STH RTN,RSAVE SAVE R15

0064R

0026R E110 SVC 1,WAIT DUMMY CALL

0062R

002AR 48F0 LH RTN,X'96' GET CURRENT PSW STATUS

0096

002ER C4F0 NHI RTN,X'BFFF' DISABLE IO BIT

BFFF

0032R 4300 B FINIS

0010R

*

0036R 40F0 ENPM STH RTN,RSAVE SAVE R15

0064R

003AR E110 SVC 1,WAIT DUMMY CALL

0062R

003ER 48F0 LH RTN,X'96' GET CURRENT PSW STATUS

0096

0042R C6F0 OHI RTN,X'0100' ADD ENABLE PM BIT

0100

PSW STATUS MODIFIER

PAGE 2

46R	4300	B	FINIS	
	0010R			
		*		
14AR	40F0	DISPM	STH	RTN.RSAVE
	0064R			SAVE R15
14ER	E110		SVC	1.WAIT
	0062R			DUMMY CALL
52R	48F0		LH	RTN.X'96'
	0096			GET CURRENT PSW STATUS
56R	C4F0		NHI	RTN.X'FEFF'
	FEFF			DISABLE PM BIT
5AR	4300	B	FINIS	
	0010R			
		*		
5ER	0004	STAT	DC	4.0
	0000			
62R	0800	WAIT	DC	X'0800'
64R		RSAVE	DS	2
66R			END	

PSW STATUS MODIFIER

0 ERRORS

* DISIOI	0022R
* DISPM	004AR
* ENIOI	0000R
* ENPM	0036R
FINIS	0010R
RSAVE	0064R
RTI	000F
STAT	005ER
WAIT	0062R

4. MODIFICATIONS TO THE TEKTRONIX PLOT-10 TERMINAL CONTROL SYSTEM FOR OPERATION AT 9600 BAUD

When operating the 4010-1 with the Terminal Control System (TCS) at data transmission rates above 4000 baud certain changes to TCS are required to eliminate the occasional appearance of short or warped vectors.

The modifications fall into two general categories:

1. Preventing the terminal from leaving the linear interpolate (line drawing) mode before the terminal has completed drawing the vector.
2. Preventing the start of a new vector before the completion of the current vector.

The following is a list of the functions which require modification at 9600 Baud.

1. Activating the bell - BEL (7)
2. Requesting dark vector - GS (29)
3. Requesting a hard copy - ESC, ETB (27, 23)
4. Bringing up the cross hair cursor - ESC, SUB (27, 26)
5. Switching from vector to alpha mode - US (31)
6. Switching to vector mode (same as 2 above) - GS (29)
7. Erasing the screen - FF (12)

4A. A Listing of the Modifications to TCS

Standard Code

Modified Code

in Subroutine BELL

SUBROUTINE BELL

SUBROUTINE BELL

C* OUTPUT (BEL)
CALL TOUTPT (7)
RETURN
END

C* OUTPUT (BEL)
CALL TOUTPT (22)
CALL TOUTPT (22)
CALL TOUTPT (7)
RETURN
END

in Subroutine TKDASH

```

      If (DTABL (I)) 21,21,22
C* OUTPUT A GS FOR A DARK VECTOR
21  CALL TOUTPT (29)
22  CALL SYCNVT
    ...
18  IF (DTABL (I)) 29,29,24
29  CALL TOUTPT (29)
    ...
11  IF (N)) 26,26,25
26  CALL TOUTPT (29)
    ...

```

```

      If (DTABL (I))21,21,22
C* OUTPUT A GAS FOR A DARK VECTOR
21  CALL TOUTPT (22)
    CALL TOUTPT (22)
    CALL TOUTPT (29)
22  CALL XYCNYT
    ...
18  IF (DTABL (I)) 29,29,24
29  CALL TOUTPT (22)
    CALL TOUTPT (22)
    CALL TOUTPT (29)
    ...
11  IF (NO) 26,26,25
    CALL TOUTPT (22)
    CALL TOUTPT (22)
26  CALL TOUTPT (29)
    ...

```

in Subroutine VECMOD

```

    ...
5  TREALX, TREALY, TIMAGX, ...
   IF (KKMODE.EQ.1)GO TO 10

```

```

    ...
5  TREALX, TREALY, TIMAGX, ...
   CALL TOUTPT (22)
   CALL TOUTPT (22)
   IF(KKMODE.EQ.1)GO TO 10
   ...

```

in Subroutine HDCOPY

```

C* OUTPUT (ESC) (ETB) to START ...
   CALL TOUTPT (27)
   CALL TOUTPT (23)
   ...

```

```

C* OUTPUT (EST)(ETB) to START ...
   CALL TOUTPT (22)
   CALL TOUTPT (22)
   CALL TOUTPT (27)
   CALL TOUTPT (23)
   ...

```

in Subroutine DCURSR

```

    ...
C* OUTPUT (ESC) (SUB) to TURN ...
   CALL TOUTPT (27)
   CALL TOUTPT (26)
   ...

```

```

    ...
C* OUTPUT (ESC) (SUB) to TURN ...
   CALL TOUTPT (22)
   CALL TOUTPT (22)
   CALL TOUTPT (27)
   CALL TOUTPT (26)
   ...

```

in Subroutine ALMODE

```

...
C* SET ALPHA MODE OUTPUT (US)
CALL TOUTPT (31)
...

```

```

...
C* SET ALPHA MODE OUTPUT (US)
CALL TOUTPT (22)
CALL TOUTPT (22)
CALL TOUTPT (31)
...

```

in Subroutine PNTMOD

```

...
C* CANCEL PREVIOUS MODES-OUTPUT(US)
CALL TOUTPT (31)
...

```

```

...
C* CANCEL PREVIOUS MODES-OUTPUT(US)
CALL TOUTPT (22)
CALL TOUTPT (22)
CALL TOUTPT (31)
...

```

in Subroutine XYCNVT

```

...
C* OUTPUT (GS) TO ENTER VECTOR MODE
CALL TOUTPT (29)
...

```

```

...
C* OUTPUT (GS) TO ENTER VECTOR MODE
CALL TOUTPT (22)
CALL TOUTPT (22)
CALL TOUTPT (29)
...

```

```

30  CALL TOUTPT(LOY)
    IF(IHOX,EQ.KPCHAR(3) GO TO 40
    KPCHAR(3) = IHOX
...

```

```

30  CALL TOUTPT(LOY)
    KPCHAR(3) = IHOX
...

```

```

41  IF(LOX,NE.KPCHAR(4) GO TO 40
    IF(KKMODE,EQ.2)GO TO 44
    IF(KMOVEF,EQ.1)GO TO 42
    IF(IFLAG,EQ.0)GO TO 42
    GO TO 100

```

```

41  IF(LOX,NE.KPCHAR(4) GO TO 38
    IF(KKMODE,EQ.2)GO TO 44
    IF(KMOVEF,EQ.1)GO TO 38
    IF(IFLAG,EQ.0)GO TO 38
    GO TO 100

```

```

40  KPCHAR(4)=LOX
42  CALL TOUTPT(LOX)
    IF(KKMODE,NE.2)GO TO 43

```

```

38  CALL TOUTPT (22)
    CALL TOUTPT(22)
40  KPCHAR(4)=LOX

```

```

44  CALL TOUTPT(LOX)
    GO TO 100
...

```

```

    CALL TOUTPT(LOX)
    IF(KKMODE,NE.2)GO TO 43
44  CALL TOUTPT (22)
    CALL TOUTPT (22)
    CALL TOUTPT (LOX)
    GO TO 100
...

```

REFERENCES

1. Newman, W.M. and Sproull, R.F., Principles of Interactive Computer Graphics. McGraw-Hill Book Co., Inc. New York. 1973.
2. 4010 and 4010-1 User's Manual, Tektronix, Inc. Document No. 070-1225-000. 1972.
3. PLOT-10 Advanced Graphing II User's Manual, Tektronix, Inc. Document No. 062-1530-00. 1973.
4. PLOT-10 Terminal Control System User's Manual, Tektronix, Inc. Document No. 062-1474-00. 1972.
5. PLOT-10 Minicomputer-40002A Interdata 3 and 4 User's Manual, Tektronix Inc. Document No. 062-1406-00. 1972.
6. Personal Communications, Roy Stehle, Stanford Research Institute, 1973.
7. Personal Communications, Will Gallant, Tektronix, Inc. 1974.

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET		1. PUBLICATION OR REPORT NO. NBSIR 74-540	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Interactive Graphics on the Sound Laboratory Data Acquisition System 2430102			5. Publication Date August 1974	
			6. Performing Organization Code	
7. AUTHOR(S) A. James Baroody, Jr.			8. Performing Organ. Report No. NBSIR 74-540	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			10. Project/Task/Work Unit No.	
			11. Contract/Grant No.	
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP) Same as No. 9			13. Type of Report & Period Covered Final	
			14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES				
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) This report describes the implementation of an interactive graphics display system on the Sound Laboratory Data Acquisition System. A brief description of the hardware and the software which supports it is presented. A detailed description of computer system-dependent programs required to support this graphics system follows.				
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons) interactive graphics; data acquisition system; graphics display terminal; minicomputer.				
18. AVAILABILITY <input type="checkbox"/> Unlimited <input checked="" type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13 <input type="checkbox"/> Order From National Technical Information Service (NTIS) Springfield, Virginia 22151			19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED	21. NO. OF PAGES 41
			20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price

